

Task#1

```
INCLUDE Irvine32.inc
.data
array1 DWORD 10,6,7,3,2
count DWORD ?
.code
main PROC

    mov ECX, LENGTHOF array1; outer i
    DEC ECX

L3:
mov ESI, OFFSET array1 ;ESI now points to the first item of array1
    mov EDI, OFFSET array1+4
    mov count,ecx ; outer
    mov ECX, LENGTHOF array1 ; inner counter j
    DEC ECX

    L1:
        mov EAX, [ESI]
        mov EBX, [EDI]
        cmp EAX,EBX
        JBE LESS
        mov [ESI],EBX
        mov [EDI],EAX

    LESS:
        add ESI, TYPE array1
        add EDI, TYPE array1
        LOOP L1

    mov ecx,count ;outer counter
    LOOP L3

    mov ECX, LENGTHOF array1;here we just print the array
    mov ESI, OFFSET array1
L2:  MOV EAX, [ESI]
    call WriteInt
    call crlf
    add ESI, TYPE array1
    LOOP L2

exit

main ENDP
END main
```

Output

```
+2  
+3  
+6  
+7  
+10  
Press any key to continue . . .
```

Task#3

```
INCLUDE Irvine32.inc  
  
Reverse proto string1: ptr byte, string2: ptr byte, size1:dword  
  
.data  
Str1 BYTE "My name is Muzamil",0  
Str2 BYTE "Hello World",0  
  
Original BYTE "Entered String: ",0  
Output BYTE "String after reversing: ",0  
  
SizeString dword ?  
  
.code  
main PROC  
    mov ecx, lengthof Str1  
    mov SizeString, ecx  
  
    mov edx, offset Original  
    call WriteString  
  
    mov edx, offset Str1  
    call WriteString  
    call crlf  
  
    invoke Reverse, addr Str1, addr Str2, SizeString  
  
    call dumpregs  
    exit  
main ENDP
```

```
Reverse proc, string1: ptr byte, string2: ptr byte, size1: dword
```

```
mov esi, string1
mov edi, string2
mov ecx, size1
```

```
mov edx, offset Output
call WriteString
```

```
mov eax, 0
mov ebx, ecx
sub ebx, 2
l1: mov al, [esi+ebx]
mov [edi], al
inc edi
dec ebx
loop l1
```

```
mov edx, offset Str2
call WriteString
call crlf
```

```
ret
Reverse endp
```

Output

```
Entered String: My name is Muzamil
String after reversing: limazuM si eman yM
```

Task#4

```

INCLUDE Irvine32.inc

.data

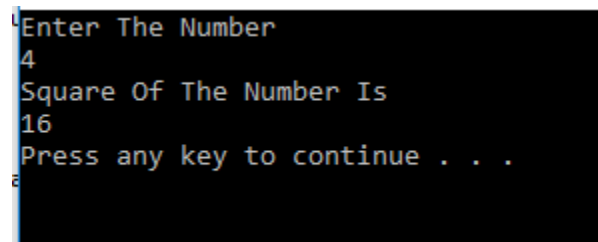
msg BYTE "Enter The Number ",0
msg2 BYTE "Square Of The Number Is ",0

.code

main PROC
call LocalSquare
exit
main ENDP
LocalSquare PROC
ENTER 0,0
mov eax,0
mov edx,0
mov edx,offset msg
call writestring
call crlf
call readdec
mov DWORD PTR[ebp-4],eax
mov eax,[ebp-4]
mul eax
mov edx,offset msg2
call writestring
call crlf
call writedec
ret
LocalSquare ENDP
END main

```

Output



```

Enter The Number
4
Square Of The Number Is
16
Press any key to continue . . .

```

Task#5

```

INCLUDE Irvine32.inc
.data
num dword 5
sqr word ?
.code
main PROC
push num
call fact
call writedec
exit
main endp
fact PROC
push ebp
mov ebp,esp
mov ecx,[ebp+8]
mov eax,1

l1:
mul ecx
loop l1
pop ebp
ret
fact endp
end main
pro PROC
push ebp
mov ebp,esp
mov eax,[ebp+16]
mov ebx,[ebp+12]
mul ebx
mov ebx,[ebp+8]
mul ebx
pop ebp

pro ENDP
call crlf
END main

```

Output

```
120Press any key to c
```

Task#6

```
INCLUDE Irvine32.inc
```

```
.data
```

```
Prime_array DWORD 20 dup(?)
array DWORD 20 dup(?)
msg1 BYTE "How many numbers you want to insert in array : ",0
msg2 BYTE "Enter the elements : ",0
maxi BYTE "The maximum Prime number is : ",0
count DWORD 0
x DWORD ?
two DWORD 2
three DWORD 3
five DWORD 5
seven DWORD 7
max DWORD ?
```

```
.code
```

```
LARGESTPRIME PROTO,len:PTR DWORD,arr:PTR DWORD
```

```
main PROC
call CheckPrime
exit
main ENDP
```

```
CheckPrime PROC
```

```
mov eax,0
mov edx,0
mov ecx,0
mov edx,offset msg1
call writestring
call readdec
mov ecx,eax
mov x,eax
```

Output

```
jmp next
label1:
add esi,4
loop l2

INVOKE LARGESTPRIME,ADDR count,ADDR Prime_array
exit

next:
mov edx,0
cmp eax,2
jne next1
mov [edi],eax
add edi,4
inc count
jmp label1

next1:
cmp eax,3
jne next2
mov [edi],eax
add edi,4
inc count
jmp label1

next2:
cmp eax,5
jne next3
mov [edi],eax
add edi,4
inc count
jmp label1

next3:
cmp eax,7
jne next4
```

% ▾

Output

Show output from: Build

```

next3:
cmp eax,7
jne next4
mov [edi],eax
add edi,4
inc count
jmp label1

```

```

next4:
mov eax,x
mov edx,0
div two
cmp edx,0
jne next5
jmp label1

```

```

next5:
mov edx,0
mov eax,x
div three
cmp edx,0
jne next6
jmp label1

```

```

next6:
mov edx,0
mov eax,x
div five
cmp edx,0
jne next7
jmp label1

```

```

next7:
mov edx,0
mov eax,x
div seven
cmp edx,0
jne next8
jmp label1

```

```

next8:
mov eax,x
mov [edi],eax
add edi,4
inc count
jmp label1

```

```

exit
CheckPrime ENDP

```

```

LARGESTPRIME PROC,len:PTR DWORD,arr:PTR

```

```

mov esi,arr
mov edi,len
mov eax,[esi]
mov ecx,[edi]
mov max,eax

```

```

l1:
mov eax,[esi]
cmp eax,max
jng NEXT1
mov max,eax
NEXT1:
add esi,4

```



```

mov esi,arr
mov edi,len
mov eax,[esi]
mov ecx,[edi]
mov max,eax

l1:
mov eax,[esi]
cmp eax,max
jng NEXT1
mov max,eax
NEXT1:
add esi,4
loop l1

call crlf
mov edx,offset maxi
call writestring
mov eax,max
call writedec
call crlf
call crlf

ret
LARGESTPRIME ENDP
END main

```

Output

```

How many numbers you want to insert in array : 3
Enter the elements : 2
Enter the elements : 3
Enter the elements : 1

The maximum Prime number is : 3

Press any key to continue . . .

```